# W-Scheduler: whale optimization for task scheduling in cloud computing

Karnam Sreenu[1] · M. Sreelatha[2]

*Abstract*—**One of the important steps in cloud computing is the task scheduling. The task scheduling process needs to schedule the tasks to the virtual machines while reducing the makespan and the cost. Number of scheduling algorithms are proposed by various researchers for scheduling the tasks in cloud computing environments. This paper proposes the task scheduling algorithm called W-Scheduler based on the multi-objective model and the whale optimization algorithm (WOA). Initially, the multi-objective model calculates the fitness value by calculating the cost function of the central processing unit (CPU) and the memory. The fitness value is calculated by adding the makespan and the budget cost function. The proposed task scheduling algorithm with the whale optimization algorithm can optimally schedule the tasks to the virtual machines while maintaining the minimum makespan and cost. Finally, we analyze the performance of the proposed W-Scheduler with the existing methods, such as PBACO, SLPSO-SA, and SPSO-SA for the evaluation metrics makespan and cost. From the experimental results, we conclude that the proposed W-Scheduler can optimally schedule the tasks to the virtual machines while having the minimum makespan of 7 and minimum average cost of 5.8.**

**Keywords** Cloud computing · Task scheduling · Multiobjective model · Whale optimization algorithm · Makespan

I. B Karnam Sreenu karnamsreenu.27@gmail.com 1 Department of Computer Science and Engineering, ANU College of Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India 2 Department of Computer Science and Engineering, RVR JC College of Engineering, Guntur, Andhra Pradesh, India

## II. Introduction

Due to the availability of big data, the requirement of cloud computing has increased in several areas, such as business and the cloud computing is the on-demand process in the recent years. Cloud computing [1,2] allows the users to access the resources, such as storage, servers, and applications from the internet [3]. The service provider of the cloud is used to manage the services, and these services are accessed by the users over the internet [4]. The cloud offers several services to the users. The most significant services are Platform as a Service (PaaS) [5], Infrastructure as a Service (IaaS) [6], Expert as a Service (EaaS) [7], and Software as a Service (SaaS) [8,9]. The users of the cloud have different jobs, and these jobs are performed simultaneously by the resources available in the cloud. The performance of the cloud computing can be improved by allocating resources to the jobs in an optimized manner. One of the critical processes of the cloud computing [10,11] is how to schedule the tasks, and the task scheduling produces great impacts on the entire cloud by affecting the Quality of Service (QoS). The task scheduling process maintains the balance among the requirements of the users and the utilization of resources. Each task requires memory, computing time and response time in different scales and the cloud computing environment has the heterogeneous resources which are distributed in the cloud geographically. The task scheduling process is affected by the above features of the cloud environment [3]. The effective task scheduling process must reduce the makespan [12] of the application. Thus, there is a need for the algorithms for scheduling tasks in the cloud which optimally allocate the tasks to the resources at the same time minimize the makespan. In cloud computing, the task scheduling process is considered as the NP-complete problem [13] where, the time required for finding the solution changes by the size of the problem [14]. Task scheduling algorithms are classified into two groups namely heuristic and meta-heuristic algorithms. Now a day, meta-heuristic algorithms are popular for task scheduling, and they find the solutions which are near to the optimal solution. Heuristic algorithms find the solution by excluding some paths in the solution space [15]. There are three types of heuristic algorithm. They are list scheduling, duplication based scheduling, and clustering based scheduling [16]. The list scheduling is performed in two steps. In the first step, it assigns the priority values to the tasks and in the second step, the tasks are allocated to the processors depends on the priority values of the tasks. In the duplication based algorithms, the identical copies of the tasks are generated which are used to decrease the application's makespan, and the duplicated copies of the tasks are allocated to the same processor [16] to reduce the cost needed for performing the computation. The clustering based algorithms groups the tasks into the cluster and the cluster of tasks are allocated to the processors. These algorithms had infinite processors and designed to work in a homogeneous system. In contrast, the meta-heuristic algorithms find the solution by utilizing the random choices [17]. The best example for the meta-heuristic algorithm is the genetic algorithm [8]. In the literature works [3,4,8,18,19], the time required for allocating the resources is increased when the number of task increases. Hence, the existing algorithms were not fitting for cloud centers with a large amount of data. The WOA [20] allows the task scheduling of the big data since it provides better performance in the unknown search space. This paper proposes the W-Scheduler for scheduling tasks to the virtual machines in the cloud computing environment. The proposed task scheduling method is based on the multiobjective model and the whale optimization algorithm. The multi-objective model calculates the cost function of CPU and memory of all the virtual machines and then, calculates the budget cost function by adding cost function of both CPU and memory. Then, it calculates the fitness by adding the makespan and the budget cost function. Based on the fitness value, it allocates the tasks to the virtual machines. The whale optimization algorithm begins with the group of random solutions. Initially, it assumes that the current solution is the best solution and performs the searching process based on the current solution. This process is repeated until the best solution reaches. The major contributions of this paper are: • The primary contribution of this work is the design of the multi-objective model for calculating the fitness value. • The secondary contribution of this work is the design of the proposed W-Scheduler based on the WOA algorithm. The proposed W-Scheduler optimally allocates the tasks to the virtual machines while maintaining the minimum makespan and minimum cost. The organization of the paper is described as follows: Sect. 2 presents the motivation of the proposed W-Scheduler for scheduling tasks to the virtual machines in the cloud environments. Section 3 describes the system model of the cloud environment. Section 4 presents our proposed task scheduling mechanism. The results and discussion are presented in Sects. 5, and 6 concludes the paper.

## III. Motivation

In this section, the various works related to the task scheduling problem in cloud computing and the challenges associated with the task scheduling are discussed.

### A. 2.1 Review of related works

The review of the research papers in the field of task scheduling in cloud environments is discussed here. HE Hua et al. [3] have suggested an adaptive multi-objective task scheduling (AMTS) approach based on particle swarm optimization (PSO) algorithm. This method optimally allocates the resources to the tasks with less time and requires average energy. The disadvantage of this method is that it did not schedule the task dynamically. Bahman Keshanchi et al. have proposed a task scheduling approach based on the genetic algorithm in [8]. Along with the genetic algorithm, this method makes use of the heterogeneous earliest finish time (HEFT) searching. The disadvantage of this method is that it takes more time to detect solutions. Xue Lin et al. [4] have proposed the task scheduling algorithm based on the dynamic voltage and frequency scaling (DVFS) technique. This method performs task scheduling with less delay and energy, and the linear-time rescheduling algorithm does the movement of the task. Xiaolong Xu et al. [18] have proposed a task scheduling method in which the resources are allocated to the tasks based on probabilistic matching (PM) and Improved Simulated Annealing (ISA). This method provides optimal allocation of resources to the tasks. Haitao Yuan et al. [19] have suggested a task scheduling method based on profit maximization algorithm (PMA). In this method, all the arrived tasks are processed in the public cloud and also in the private cloud. The difficulty in the profit maximization algorithm was removed by the simulated annealing particle swarm optimization algorithm (SAPSO). This method did not work on the real cloud environments. Yibin Li et al. [21] have suggested an Energy-aware Dynamic Task Scheduling (EDTS) algorithm based on the Dynamic Voltage Scaling (DVS) technique, and also they have proposed a Critical Path Assignment (CPA) algorithm for finding the critical paths. The advantage of this method is that it had greater efficiency. The drawback of this method is that it was designed for only Android devices and needs improvements. Zhifeng Zhong et al. [22] have presented task scheduling algorithm based on Greedy Particle Swarm Optimization (GPSO). It had some advantages, such as better convergence rate, balanced workload. This method considered only the task size and the virtual machines' ability but did not consider the other factors, such as bandwidth. Chunling Cheng et al. [23] have suggested a task scheduling method based on the Vacation Queuing Theory for minimizing the need for energy. The disadvantage of this method is that the performance of the task scheduling was low. Hongyan Cui et al. [24] presented the task scheduling in the cloud environment based on the Markov model. Various objects such as reliability, makespan, and flow time found for the scheduling is found as the optimization problem. They have proposed the Genetic Algorithm-based Chaotic Ant Swarm (GA-CAS) algorithm for scheduling the tasks. This model has an improved rate of convergence. Sanjaya K. Panda et al. [25] proposed the task scheduling for the cloud platforms by considering the factor task allocation. The proposed algorithm depends on the Min–Min and Max–Min algorithm to make it suitable for the multi-cloud environment. Factors such as transfer cost and time stamp in the task scheduling were not discussed in this work.

## IV. 2.2 Challenges

One of the critical problems in cloud computing is the scheduling of tasks to the resources. The existing task scheduling algorithms did not solve the various crucial factors associated with the task scheduling problem. The time required for allocating the resources was increased when the number of task increases. Hence, the existing algorithms were not fitting for cloud centers with a large amount of data. Variation in tasks and time adjustment are the major challenges during the process of task scheduling. Most of the existing systems perform task scheduling by adjusting the tasks. They did not evaluate the time overhead. In some situations, the estimated workload of the system was smaller than the actual workloads of the system which leads to performance degradation of the task scheduling algorithms. The sudden oscillation in the estimation of workload also influences the stability of the system [18].

## V. 3 System model

This section describes the system model of the proposed task scheduling mechanism in the cloud environment. Figure 1 represents the system model of task scheduling mechanism in the cloud computing. The task manager collects the tasks from the various users. The users submit the task requests to the task manager. The task manager manages the database to store every user request. The task manager organizes the user tasks and provides the status of the task to the user. The task manager contains the information about the status of the virtual machine. The task manager provides these task requests to the task scheduler. Task scheduler is a device which provides the priority to the incoming tasks. The task scheduler analyzes the memory requirement, cost, deadline, and the required budget of the tasks. The cloud environment contains many physical machines. The virtual machine present in the physical machine can process many tasks. The task scheduler allocates tasks to the virtual machines presented in the cloud environment. Assume that, the cloud which consists of 100 physical machines and each physical machine consists of 10 virtual machines. This can be represented as, Cloud,C = P1, P2 ..., P100 (1) where, C represents the cloud and P1, P2 ..., P100 represents the physical machines presented in the cloud. The following equation can represent the physical machine P1. P1 = V1, V2 ..., Vj ..., V10 (2) where, V1, V2 ..., Vj ..., V10 represents the virtual machines presented in the physical machine P1.Each virtual machine has the central processing unit (CPU) and the memory. Here, the capacity of the CPU is 1860 MIPS (Millions Instructions per second) or 2660 MIPS. That is, the CPU can be able to perform 1860 or 2660 millions of instructions in one second. The size of the memory is 4 GB. The total number of tasks is 100, and each task has the different user cost of CPU, the user cost of memory, deadline, and budget cost. T ask = T1, T2 ..., Ti ..., T100 (3) where, T1, T2 are the first and second tasks respectively. Tirepresents the i th task and T100 represents the 100 th task. Figure 2 explains the operation of the task scheduler. The task manager provides the tasks to the scheduler. Each task provided by the user contains the following parameters, memory requirement, cost, deadline, and the required budget. Based on this parameter the task scheduler prioritizes the task and schedules it accordingly. In Fig. 2, the memory requirement, cost, deadline, and the required budget of the task Ti are analyzed by the scheduler and provided accordingly to the virtual machine for processing. In Fig. 2, the term CU i represents the cost of CPU of the task Ti which was defined by the users, MU i is the cost of the memory of task Ti defined by the users, DU i represents the deadline for the task Ti , and BU i is the budget cost of Ti .

## VI. 4 Proposed task scheduling method based on whale optimization algorithm

This section describes the proposed task scheduling method for scheduling tasks to the virtual machines in the cloud computing environments. The proposed task scheduling method is based on the multi-objective model [26] and the whale optimization algorithm [20]. Figure 3 shows the block diagram of the proposed W-Scheduler for scheduling tasks to the virtual machines in the cloud environments. The multi-objective model calculates the cost function of CPU and memory of all the virtual machines and then, calculates the budget cost function by adding cost function of both CPU and memory. Then, it calculates the fitness by adding the makespan and the budget cost function. Based on the fitness value, it allocates the tasks to the virtual machines. The whale optimization algorithm begins with the group of random solutions. Initially, it assumes that the current solution is the best solution and performs the searching process based on the current solution. This process is repeated until the best solution reaches. The objective of the task scheduling is to optimally sched ule the tasks to the resources while obtaining the minimum makespan and minimum budget cost. Makespan represents the total time needed for executing all the tasks.

## VII. 4.1 Multi-objective model for task scheduling

The multi-objective model [26] for scheduling the tasks to the virtual machines is described here. At first, the multiobjective model calculates the cost function of CPU and memory of all the virtual machines presented in the cloud environment and calculates the budget cost. The budget cost function is calculated by adding the cost function of CPU and memory. Then, the fitness is calculated by combining the budget cost function and the makespan of the scheduling process.

### *A. 4.1.1 Fitness calculation*

The fitness is calculated to find the quality of the optimal solutions, and the solutions must have minimum makespan and minimum cost function. At first, the cost function of CPU and memory is calculated. The following equations can calculate the cost functions of CPU and memory of the virtual machine Vj . C (x) = — V M— j=1 Ccost (j) (4) where, Ccost (j)is the cost of the CPU of the virtual machine Vj , —V M—represents the total number of virtual machines. Then, the Ccost (j) is calculated as follows, Ccost (j) = Cbase × Cj × ti j + CTrans (5) where, Cbase is the base cost, Cj represents the CPU of the virtual machine Vj , and ti jrepresents the time in which the task Ti is processed in the resource Rj .CTrans is the transmission cost of the CPU. Here, Cbase and CTrans are constant. Cbase = 0.17/hr (6) CTrans = 0.005 (7) The cost function of the memory is calculated by, M (x) = — V M— j=1 Mcost (j) (8) where, Mcost (j) represents the cost of the memory of the virtual machine Vj , —V M—represents the total number of virtual machines. Then, Mcost (j) is calculated as follows, Mcost (j) = Mbase × Mj × ti j + MTrans (9) where, Mbase represents the base cost of the memory, Mjrepresents the memory of the virtual machine Vj , and ti jrepresents the time in which the task Ti is processed in the

BibTeX does not work by magic. It doesn't get the bibliographic data from thin air but from .bib files. If you use BibTeX to produce a bibliography you must send the .bib files.

LaTeX can't read your mind. If you assign the same label to a subsubsection and a table, you might find that Table I has been cross referenced as Table IV-B3.

LaTeX does not have precognitive abilities. If you put a `\label` command before the command that updates the counter it's supposed to be using, the label will pick up the last counter to be cross referenced instead. In particular, a `\label` command should not go before the caption of a figure or a table.

Do not use `\nonumber` inside the `{array}` environment. It will not stop equation numbers inside `{array}` (there won't be any anyway) and it might stop a wanted equation number in the surrounding equation.

### *B. Some Common Mistakes*

- The word "data" is plural, not singular.
- The subscript for the permeability of vacuum $\mu_0$, and other common scientific constants, is zero with subscript formatting, not a lowercase letter "o".
- In American English, commas, semicolons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
- A graph within a graph is an "inset", not an "insert". The word alternatively is preferred to the word "alternately" (unless you really mean something that alternates).
- Do not use the word "essentially" to mean "approximately" or "effectively".
- In your paper title, if the words "that uses" can accurately replace the word "using", capitalize the "u"; if not, keep using lower-cased.
- Be aware of the different meanings of the homophones "affect" and "effect", "complement" and "compliment", "discreet" and "discrete", "principal" and "principle".
- Do not confuse "imply" and "infer".
- The prefix "non" is not a word; it should be joined to the word it modifies, usually without a hyphen.
- There is no period after the "et" in the Latin abbreviation "et al.".
- The abbreviation "i.e." means "that is", and the abbreviation "e.g." means "for example".

An excellent style manual for science writers is [7].

### *C. Authors and Affiliations*

**The class file is designed for, but not limited to, six authors.** A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

### *D. Identify the Headings*

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is "Heading 5". Use "figure caption" for your Figure captions, and "table head" for your table title. Run-in heads, such as "Abstract", will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced.

### *E. Figures and Tables*

*a) Positioning Figures and Tables:* Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation "Fig. 1", even at the beginning of a sentence.

TABLE I
TABLE TYPE STYLES

| **Table Head** | **Table Column Head** | | |
|---|---|---|---|
| | ***Table column subhead*** | ***Subhead*** | ***Subhead*** |
| copy | More table copy[a] | | |

[a]Sample of a Table footnote.

Fig. 1. Example of a figure caption.

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity "Magnetization", or "Magnetization, M", not just "M". If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write "Magnetization (A/m)" or "Magnetization {A[m(1)]}", not just "A/m". Do not label axes with a ratio of quantities and units. For example, write "Temperature (K)", not "Temperature/K".

## ACKNOWLEDGMENT

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g". Avoid the stilted expression "one of us (R. B. G.) thanks . . .". Instead, try "R. B. G. thanks. . .". Put sponsor acknowledgments in the unnumbered footnote on the first page.

## REFERENCES

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first . . ."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

## REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.
[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[4] K. Elissa, "Title of paper if known," unpublished.
[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.